REMARKS

Claims 1, 3-5, 7-10, 12, 13, 15, 16, 18-20, 22, 23, 25-27, 29-33, and 35 are all the claims pending in the application. The claims have been amended pursuant to a telephone conference with the Examiner on October 24, 2006.

Claims 1, 3, 5, 7-10, 12, 15, 16, 18, 20, 22-23, 25, 27, 29-31, 33, and 35 stand rejected under 35 U.S.C. §102(e) as being anticipated by Scholl, et al., hereinafter referred to as "Scholl" (6,145,001), in view of Gupta, et al., hereinafter referred to as "Gupta" (6,457,173). Claims 4, 13, 19, 26, and 32 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Scholl, in view of Gupta, and Rogers, et al., hereinafter referred to as "Rogers" (6,094,655).

A. The Rejection Based on Scholl and Gupta.

The invention comprises a method for processing the same request from a client program to multiple instances of the same server program over the same protocol. Scholl discloses a system that divides (parses) a client's Web request into different portions, sends the different portions of the divided request to dissimilar managed networks using different protocols (not the same request applied to multiple instances of the same server program over the same protocol, as required by Applicant's claims) and then combines results obtained from the dissimilar managed networks into a single unified result that appears as a single Web file to the user (Abstract and flowchart in Figure 6 of Scholl). Gupta discloses a method of designing instruction formats for performing parallel processing within a processor, such as an application specific instruction-set processor (column 1, lines 5-50) which is completely unrelated to the claimed processing of the same request from a client program to server programs.

The claim language provides "a method for processing the same request from a client program to multiple instances of the same server program over the same protocol." The specific claim language in question provides "generating a plurality of request instances of said same request . . . specifying target instances of said same server program to form a fan out target list . . . said client program, said same server program, and said

same protocol are not modified" which is fundamentally different than generating a plurality of requests as proposed in the Office Action.

The claims require "processing the same request from a client program to multiple instances of the same server program over the same protocol" which is something that Scholl does not do because Scholl divides the request and then processes the divided portions of the request using dissimilar managed networks and devices (Abstract, lines 6-9 and 14-17; and flowchart in Figure 6, especially items 22 and 27). The claims define "a method for processing the same request from a client program to multiple instances of the same server program over the same protocol" and "generating a plurality of request instances of said same request . . . specifying target instances of said same server program to form a fan out target list . . . said client program, said same server program, and said same protocol are not modified."

With respect to Gupta, the disclosure of a multiplexer as a functional unit within an integrated circuit processor and the teaching of performing parallel operations within a processor relates do not teach or suggest the claimed "method for processing the same request from a client program to multiple instances of the same server program over the same protocol" much less the claimed process of "generating a plurality of request instances of said same request . . . specifying target instances of said same server program to form a fan out target list . . . said client program, said same server program, and said same protocol are not modified" for which Gupta is referenced as teaching.

More specifically, the multiplexer mentioned in column 4, line 9 of Gupta is merely described as one of the functional units that would be found within a processor, such as an integrated circuit processor. There is no indication within Gupta that the multiplexer would be used in any way for "generating a plurality of request instances of said same request . . . specifying target instances of said same server program to form a fan out target list . . . said client program, said same server program, and said same protocol are not modified." Further, the teaching of operation sets and operation groups appearing in column 7, lines 6-31 of Gupta relates to the different logical operations that a processor may perform. See column 7, lines 2-6, where Gupta explains that the operation sets and operation groups being discussed relate to parallel instruction computing architectures that will be utilized in processors.

By grouping similar operations together into sets and groups, the process of designing the architecture of a processor is made more efficient. For example, column 7, lines 18-20 of Gupta explain that is more efficient to group arithmetic operations into an operation set. There is nothing within Gupta or any other prior art reference of record which relates such teachings to the claimed "method for processing the same request from a client program to multiple instances of the same server program over the same protocol." Gupta relates to designing the architecture of processors and not to processing requests between clients and servers.

The reader is directed to column 6, lines 25-31 of Scholl which explains that Scholl "parses [divides] and translates the request, converts [makes the divided requests different] the request into the appropriate network management service requests, and forwards each request to the appropriate managed network 6 using the appropriate communication protocol" (bracketed material added). These concepts are repeated at many points within Scholl (see, for example, column 7, lines 2-5; column 7, lines 58-66; Abstract). The parsing of the single request into multiple requests indicates that Scholl clearly divides the request into multiple sub-requests that are different from one another because each sub-request includes a different portion of the original request. Further, by using the word "appropriate" before managed network and communication protocol, Scholl indicates that the managed networks and protocols are different. Therefore, Scholl cannot teach or suggest "a method for processing the same request from a client program to multiple instances of the same server program over the same protocol" as in the claimed invention.

With the conventional systems, the client program, the protocol and/or the server programs are modified to allow a request from the client program to be issued to multiple instances of the server program. This process is very time consuming and expensive. The invention is fundamentally different because the invention only modifies the request (and the response to the request) to accommodate the different instances of the server program. By modifying only the request and the response, the invention avoids the need to modify the server program, client program, or protocol. Further, because only the request and the response are modified, the process can be performed automatically using the invention, which makes the request and response transparent (e.g., appear as a one-to-one communication) to the server and client programs.

A single client program often needs to send the same request to several instances of a server program and process the responses obtained from each instance of the program. Each server is assumed to be executing the same program. However, the different instances of the server program have different data and/or states. Conventionally, the client program would have to be customized to accommodate each different instance of the server program. The invention eliminates the need to customize the client program and automates the communication without modification of the programs or the protocol.

As shown in Applicants' Figure 2A, the invention comprises a multiplexor M, with extensions to handle specific protocols. Further, the invention permits context-free operation under certain assumptions, allowing a single multiplexor to handle communications between many pairs of C-S, C'-S', C"-S" that share the same protocol P, without any additional programming. As shown in Applicants' Figure 2B, the inventive multiplexor 22 can operate with different instances of the client program C, C' and different instances of the server program S0, S1, S2, S0', S1', etc., as well as operate with multiple instances of both programs simultaneously.

Applicants note that Scholl discloses a system that parses the client request into multiple different requests where each requested is submitted to a different managed network, while the claimed invention presents a method of processing multiple instances of a server program based on the same request from a single client program. Therefore, it is Applicants' position that Scholl is fundamentally different than the claimed invention and does not describe a similar or equivalent process as in the claimed invention.

The invention generates multiple instances of the same request from the single original request sent by the client program. The servers will send back a response to the client program, either indicating an error condition or successful execution, possibly returning some data. These responses are then modified and combined by the intermediary to correspond to the protocol instance of the client program (e.g., to the same format, version, data structure, etc. of the original request) so that the client program believes it is talking to a single server program in a one-to-one communication environment.

This is fundamentally different than what is being described in Scholl, because Scholl only directs a different portion of any client request to a single server. More

specifically, item 25 in Figure 6 illustrates that Scholl merely forwards the portion of the same request to the appropriate network management proxy agent. Scholl does not generate instances of the same request, and instead merely sends the portion of the request to the server that will supply the appropriate answer.

More specifically, in column 7, line 58-column 8, line 14, Scholl explains that the request is parsed and translated with a programmable device, or a circuit device, into at least one network management request ("NMR"). The request is analyzed as to whether processing the request requires interaction with a managed network. If not, the request is processed locally; and if so, the request is forwarded to an appropriate network management proxy agent 25. After the forwarding step 25, the network management proxy agent determines whether the information is in the local database. If yes, the information is obtained therefrom; and if not the request is transmitted to a managed network by access protocols. Then network management information transmissions are received in response to each request from a managed network (and may be stored in the local database for future retrieval).

This demonstrates that Scholl does not generate "a plurality of request instances of said same request using said intermediary" (independent claims 1, 16, 23, and 30) or modify "said same request to create multiple request instances of said same request" and transfer "said request instances of said same request to corresponding ones of said instances of said same server program" (independent claim 8) as in the claimed invention.

While Scholl states that the request is analyzed as to whether processing the request requires interaction with one or more managed networks, this does not indicate that multiple instances of the same request are transferred to different instances of the same server program, as in the claimed invention. To the contrary, the system disclosed in Scholl merely determines which single managed network will contain information that responds to the request (or a portion of the request) and then makes that request (or portion of the request) to that given network. There is no disclosure in Scholl that would teach or suggest to one ordinarily skilled in the art to generate "a plurality of request instances of said same request . . . specifying target instances of said same server program to form a fan out target list . . . said client program, said same server program, and said same protocol are not modified" as in the claimed invention. Instead, Scholl merely

requests that each managed network retrieve its portion of the information needed to respond to the request.

Therefore, as shown above, Scholl discloses a system that parses the client request into multiple different requests where each requested is submitted to a different managed network, while the claimed invention presents a method of processing multiple instances of the same request from a single client program. Gupta relates to designing the architecture of processors and not to processing requests between clients and servers. Therefore, it is Applicants position that Scholl and Gupta are fundamentally different than the claimed invention. Thus, Applicants submit that the proposed combination of Scholl and Gupta does not teach or suggest "generating a plurality of request instances of said same request using said intermediary . . . specifying target instances of said same server program to form a fan out target list . . . said client program, said same server program, and said same protocol are not modified " (independent claims 1, 16, and 23) or "modifying said same request to create multiple request instances of said same request" and "transferring said request instances of said same request to corresponding ones of said instances of said same server program" (independent claim 8) as in the claimed invention. In view of the foregoing, Applicants submit that independent claims 1, 8, 16, 23, and 30 are not rendered obvious by Scholl and Gupta and are patentable. Further, dependent claims 3, 5, 7, 9, 10, 12, 15, 18, 20, 22, 25, 27, 29-31, 33, and 35 are similarly patentable, not only by virtue of their dependency from a patentable independent claim, but also by virtue of the additional features of the invention defined. In view the forgoing, the Examiner is respectfully requested to reconsider and withdraw this rejection.

B. The Rejection Based on Scholl in view of Rogers

Rogers is referenced for the limited purpose of teaching specific operations that are performed on the response data including listing, adding, subsets, maximums, minimums, and averages. Rogers is not referenced (and does not teach or suggest) the inventive feature of processing multiple instances of the same request to different instances of the same server program as in the claimed invention (as explained above). Therefore, Rogers does not cure the deficiencies of Scholl and Gupta discussed above

with respect to independent claims 1, 8, 16, 23, and 30 and such independent claims are patentable over any combination of Scholl and Rogers. Thus, it is Applicants position that independent claims 1, 8, 16, 23, and 30 are patentable over the prior art of record. Further, dependent claims 4, 13, 19, 26, and 32 are similarly patentable, not only because they depend from a patentable independent claim, but also because of the additional features the dependent claims define. In view the foregoing, the Examiner is respectfully requested to reconsider and withdraw this rejection.

B. Formal Matters and Conclusion

In view of the foregoing, Applicants submit that claims 1, 3-5, 7-10, 12, 13, 15, 16, 18-20, 22, 23, 25-27, 29-33, and 35, all the claims presently pending in the application, are patentably distinct from the prior art of record and are in condition for allowance. The Examiner is respectfully requested to pass the above application to issue at the earliest possible time.

Should the Examiner find the application to be other than in condition for allowance, the Examiner is requested to contact the undersigned at the local telephone number listed below to discuss any other changes deemed necessary.

Please charge any deficiencies and credit any overpayments to Attorney's Deposit Account Number 09-0441.

Respectfully submitted,

_____

Frederick W. Gibb, III
Registration No. 37,629

Date: _____

Gibb IP Law, LLC
2568-A Riva Road, Suite 304
Annapolis, Maryland 21401
(301) 261-8071
Customer No. 29154